

# **PYTHON**

## **DESCOMPLICADO**



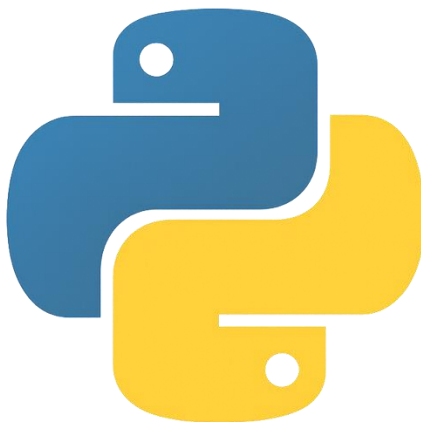
**Aprenda seletores  
com exemplos simples  
e fáceis de entender**

# Dominando Seletores

## Introdução

Se você está começando com Python, provavelmente já esbarrou na dúvida: “como eu pego só uma parte do texto?” ou “como acesso apenas um item da lista?”.

A boa notícia é que o Python tem uma forma super prática de fazer isso: seletores. Eles funcionam como uma lupa que aponta exatamente para o pedaço de dado que você quer.



# O que são Seletores?

Pense nos seletores como o controle remoto da sua série favorita: com ele, você vai direto para o episódio ou cena que deseja.

No Python, é a mesma coisa: eles servem para escolher pedaços específicos de textos, listas, dicionários e até conjuntos.

Exemplo:

```
nome = "Python"
print(nome[0])    # Resultado: P
print(nome[-1])   # Resultado: n
```

# Seletores em Strings

Strings são sequências de caracteres, e isso significa que podemos tratá-las quase como listas de letras.

O legal é que dá para pegar só um pedaço do texto, pular caracteres ou até mesmo ler de trás para frente.

```
texto = "Programar é divertido!"
print(texto[0])      # P (primeiro caractere)
print(texto[3:9])    # gram (do índice 3 até 8)
print(texto[:10])    # Programar é (do início até o índice 9)
print(texto[::2])     # Pormrédvrt! (pula de 2 em 2)
print(texto[::-1])   # !odotrevid é ramargorP (inverte a string)
```

**Dica:** `[::-1]` é um truque muito usado para inverter textos.

# Seletores em Listas

As listas guardam elementos em ordem definida, e por isso podemos usar seletores de várias formas.

Além de acessar, podemos também modificar, adicionar ou remover itens.

```
numeros = [10, 20, 30, 40, 50]
print(numeros[0])    # 10
print(numeros[1:4])  # [20, 30, 40]
print(numeros[-2])   # 40
```

E olha só: listas podem ser modificadas!

```
numeros[2] = 99
print(numeros)    # [10, 20, 99, 40, 50]
```

# Seletores em Tuplas

As tuplas parecem listas, mas com uma diferença fundamental: não podem ser alteradas.

Elas são úteis quando você quer garantir que os dados fiquem fixos.

```
cores = ("azul", "verde", "vermelho", "amarelo")
print(cores[0])      # azul
print(cores[-1])     # amarelo
print(cores[1:3])    # ('verde', 'vermelho')
```

# Seletores em Dicionários

Um dicionário é como uma agenda de contatos: você procura pelo nome (chave) e encontra o número (valor).

Aqui não usamos índices, mas sim as chaves para acessar os dados.

```
peessoa = {"nome": "Ana", "idade": 25}
print(peessoa["nome"])      # Ana
print(peessoa.get("idade")) # 25
```



# Seletores em Conjuntos (Sets)

Os conjuntos (sets) não mantêm ordem e não aceitam itens repetidos.

Por isso, não conseguimos acessar por índice, mas podemos verificar se algo está lá dentro.

```
frutas = {"maçã", "banana", "laranja", "maçã"}
print(frutas)           # {'banana', 'laranja', 'maçã'} (sem duplicados)
print("banana" in frutas) # True
print("uva" in frutas)   # False
```



# Obrigada por ler até aqui!

Esse ebook foi gerado por IA e diagramado por Sarah Banko.

O intuito do ebook é a aplicação prática dos conhecimentos adquiridos no curso Fundamentos de IA Generativa.

